Deep photo style transfer

Presentation by Mukul Ranjan as part of IITG.ai

Link to original paper: https://arxiv.org/pdf/1703.07511.pdf
Link to code: https://github.com/luanfujun/deep-photo-styletransfer

Introduction

- The paper deals with transforming the input photos in such a way that the transfer is free of any distortion and yields satisfying photorealistic style transfers in a broad variety of scenarios, including transfer of the time of day, weather, season, and artistic edits.

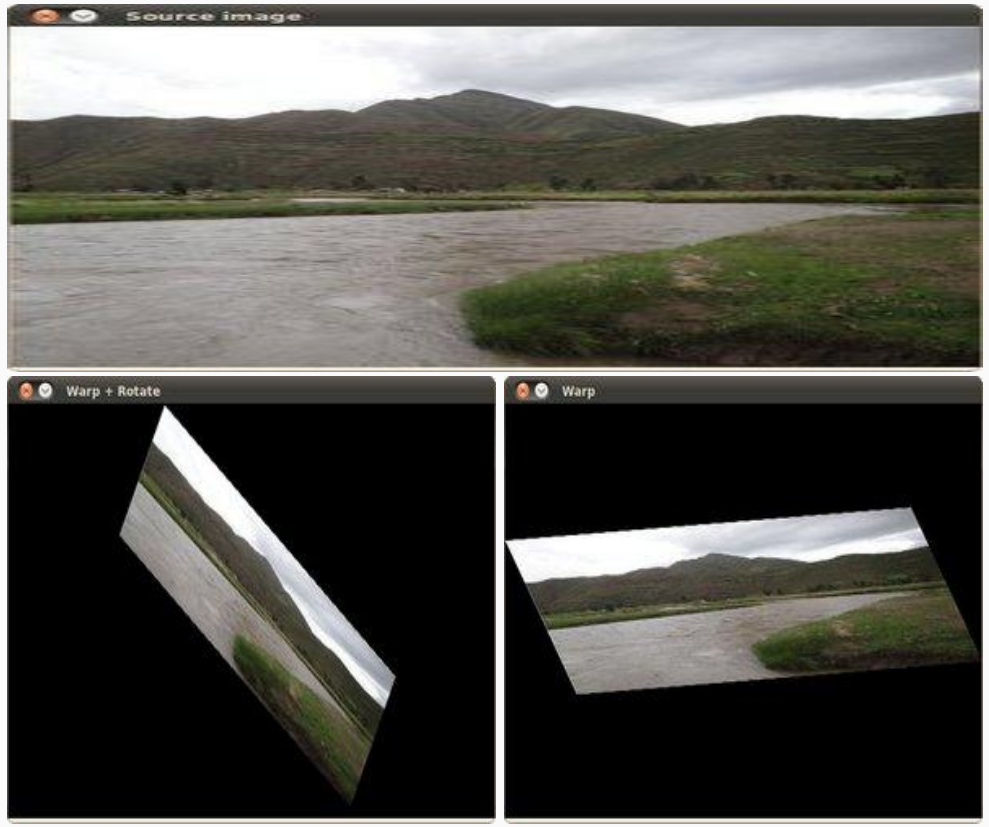
Overview of the algorithm:

Neural Style Transfer (NST) + Photorealism Regularization* + Semantic Segmentation* = Deep Photo Style Transfer

Mathematics in the paper

Affine transformation:

- A transformation of the form A(**v**) = A.**v** + **b**, where 'A' is a matrix representing linear transformation and **b** ia vector. In other word it is *linear transformation with translation*.
- Every linear transformation is affine (just set **b** to the zero vector). However, not every affine transformation is linear.
- Affine transformation makes sure that points map to points, line map to lines and planes map to planes. Also, line segments map to line segments.One nice consequence of this fact is that one can calculate the image of a polygon by simply computing the images of its vertices.
- This also makes sure that **'Parallelism is Preserved'** i.e. *parallelograms map to parallelograms and ellipses map to ellipses*.
- Useful links to understand it: [1],[2]



Mathematics in the paper

Matting:

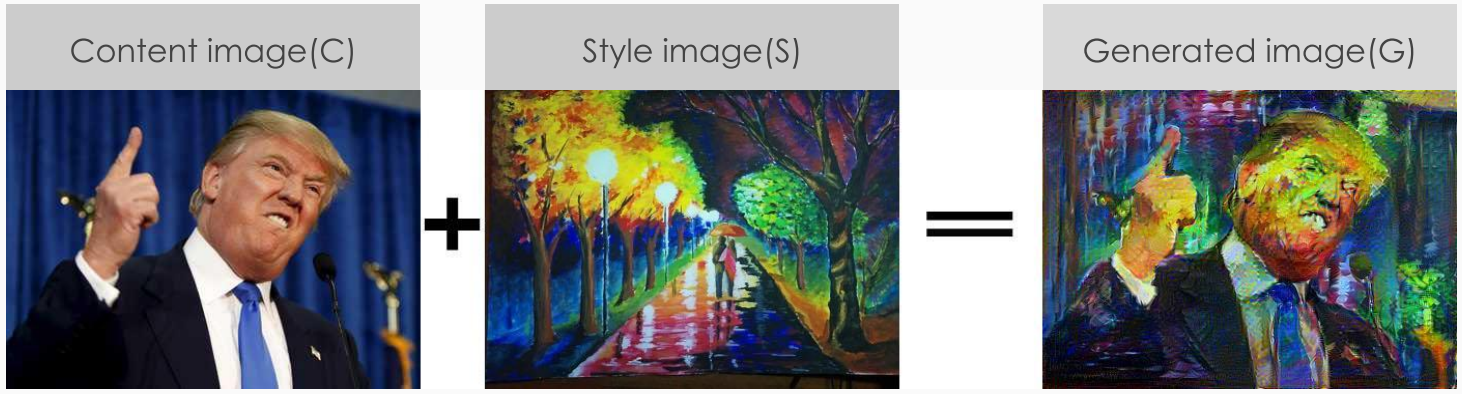
- Matting refers to the process of extracting foreground object from an image.
- Matting tasks usually produces a **"matte"** that can be used to separate foreground from the background in a given image.
- Matte can also used to combine a given foreground on a different background to produce new plausible image.

- **Excited?** Read [this]
- **Over excited?** Read [this]

Source	Matte	Separated image
		

Neural Style Transfer (NST)

- This is the method on which deep photo style transfer is based.
- Neural Style Transfer as shown below merges two images, namely, "content image(C)" and "style image (S)" to produce "generated image(G)".
- This is the method on which deep photo style transfer is based.
- Neural Style Transfer as shown below merges two images, namely, "content image(C)" and "style image (S)" to produce "generated image(G)".



How does NST works?

Goal:

To have a "generated image" G looks like "content image" C and to have the artistic effect of style image S.

How it is done:

- Take a pretrained model and take the activations of one of its layer to denote the content image. Since pretrained models are good at detecting the things correctly so we can achieve our target of getting generated image same as that of content image if we minimize the difference between the each pixel value of content image(input image) activation and generated image activation.
- DPST paper denotes the activation of a layer l by function F[O] for output or generated image and F[**I**] for input or content image C.
- Content loss(the difference in ***"content"*** between the content reference image and our generated image) can be given by following equation:

$$\mathcal{L}_c^{\ell} = \frac{1}{2N_{\ell}D_{\ell}} \sum_{ij} (F_{\ell}[O] - F_{\ell}[I])_{ij}^2$$

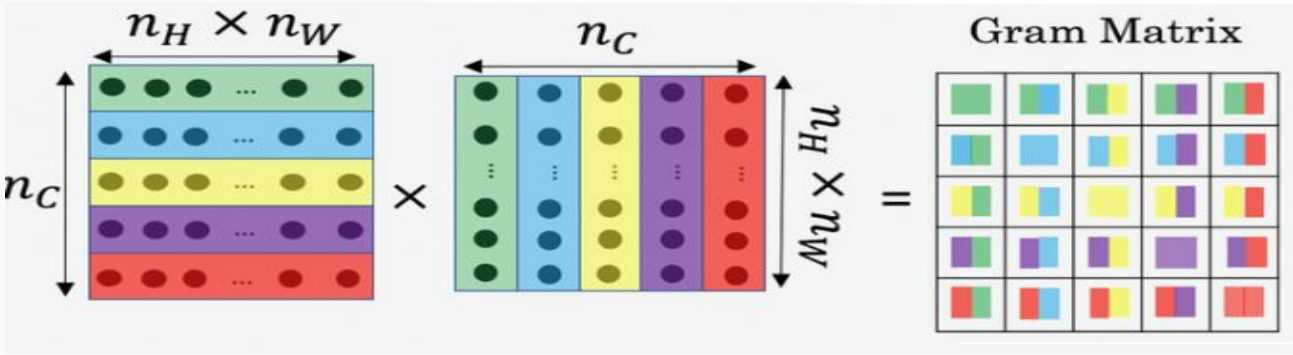
How to get the style: Gram Matrix and style loss

Gram matrix or Style matrix:

- In linear algebra gram matrix is define as inner product or the dot product of two vectors.
- In this context, gram matrix is used as style matrix. So the gram matrix is the dot product of activations matrix (a conv layer will give us a matrix of shape (n_c, n_h, n_w)) of the layer l with the shape of $N_c \times N_c$ (see the image), where N_c denotes the number of filter in lth layer.
- Why Gram Matrix works is another problem, if you are excited then look at Demystifying Neural Style Transfer paper, this blog and this quora answer.

$$G^l_{i,k} = \sum_k F^l_{i,k} F^l_{j,k}$$

Gram Matrix Equation



How to get the style: Gram Matrix and style loss

Style loss:

- It is defined as the difference in "style" between the style reference image and our generated image. It is here, in the style loss where we use the Gram matrix. The style loss is the normalized, squared difference in Gram matrices between the two images.
- So here style is nothing but the gram matrix of the activations of the style image.
- The equations for the style loss can be given as follows(Here Gl[O] and Gl[S] represent the gram matrix of the activation of lth layer for output and style image respectively):

$$\mathcal{L}_s^\ell = \frac{1}{2N_\ell^2} \sum_{ij} (G_\ell[O] - G_\ell[S])_{ij}^2$$

Total Loss

Total Loss:

Total loss is the sum of style loss and content loss given by following equation(α, β and Γ are hyperparameters and L is the total number of layers):

$$\mathcal{L}_{\text{total}} = \sum_{\ell=1}^{L} \alpha_\ell \mathcal{L}_c^\ell + \Gamma \sum_{\ell=1}^{L} \beta_\ell \mathcal{L}_s^\ell$$

Overcoming the problems of NST: Photorealism Regularization

- The goal is to transfer the style of the reference to the input while keeping the result photo-realistic
- Finally building the model on the matting laplacian help to overcome the problems of distortion. The regularization term can be given as :

$$\mathcal{L}_m = \sum_{c=1}^{3} V_c[O]^T M_I V_c[O]$$

- Here $V_c[O]$ (N x 1)represent the vectorized version of output image O and the matrix M_I (N x N) depends only on the input image.

If output pixel is denoted by x, then the equation on the left can be written as (LHS of both equation represent same term):

$$\mathcal{L}_{\text{tv}}(\vec{x}) = \sum_{c=1}^{3} \sum_{i=1}^{H-1} \sum_{j=1}^{W-1} (x_{i,j+1,c} - x_{i,j,c})^2 + (x_{i+1,j,c} - x_{i,j,c})^2$$

Overcoming the problems of NST: Style loss with semantic segmentation

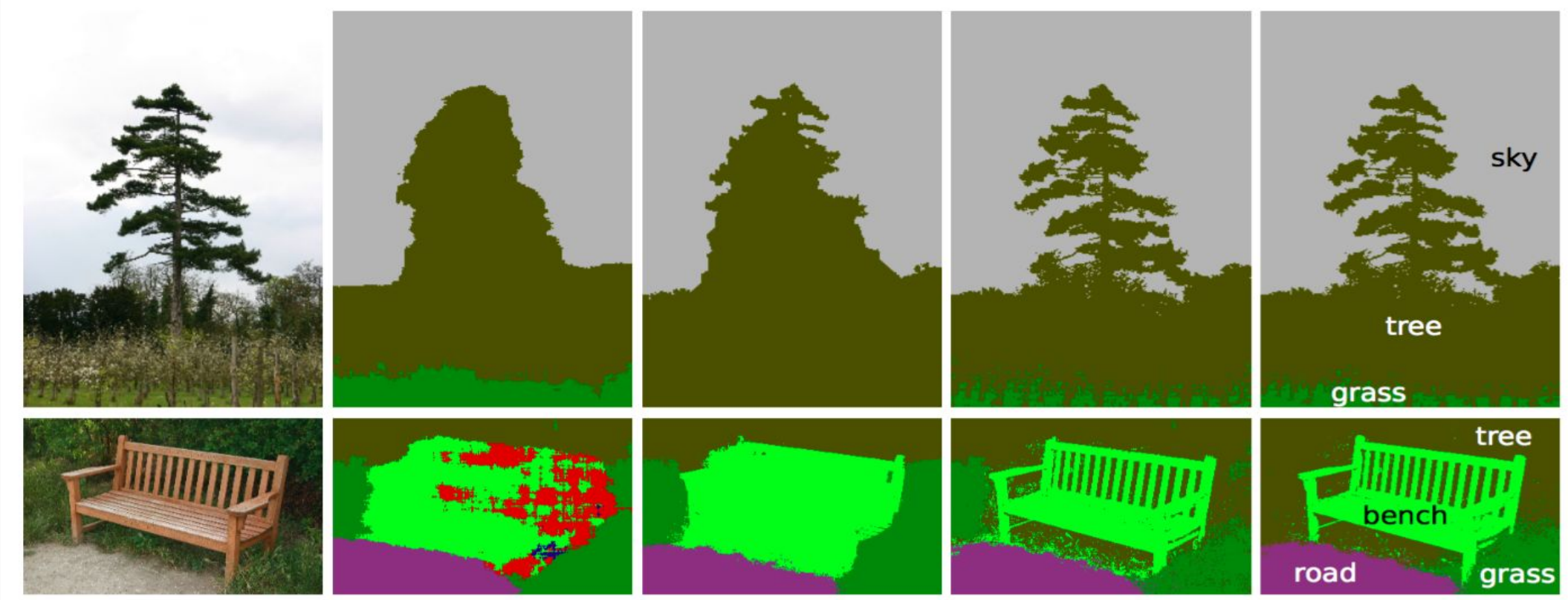
- Photo-realistic regularization make sure that style is transferred without distorting the image, for example windows stayed aligned on a grid. But to map appropriate part of the input with style image, we need to use semantic segmentation. For example to make sure that building should be mapped with the building not with the sky.

- Semantic segmentation is understanding an image at pixel level i.e, we want to assign each pixel in the image an object class.

- Semantic segmentation is build upon the 'Dilated Net'(link). Image segmentation masks are generated for the inputs for a set of common labels (sky, buildings, water, and so on), and masks are added to the input image as additional channels.

- Style loss is updated accordingly as shown in the next slides.

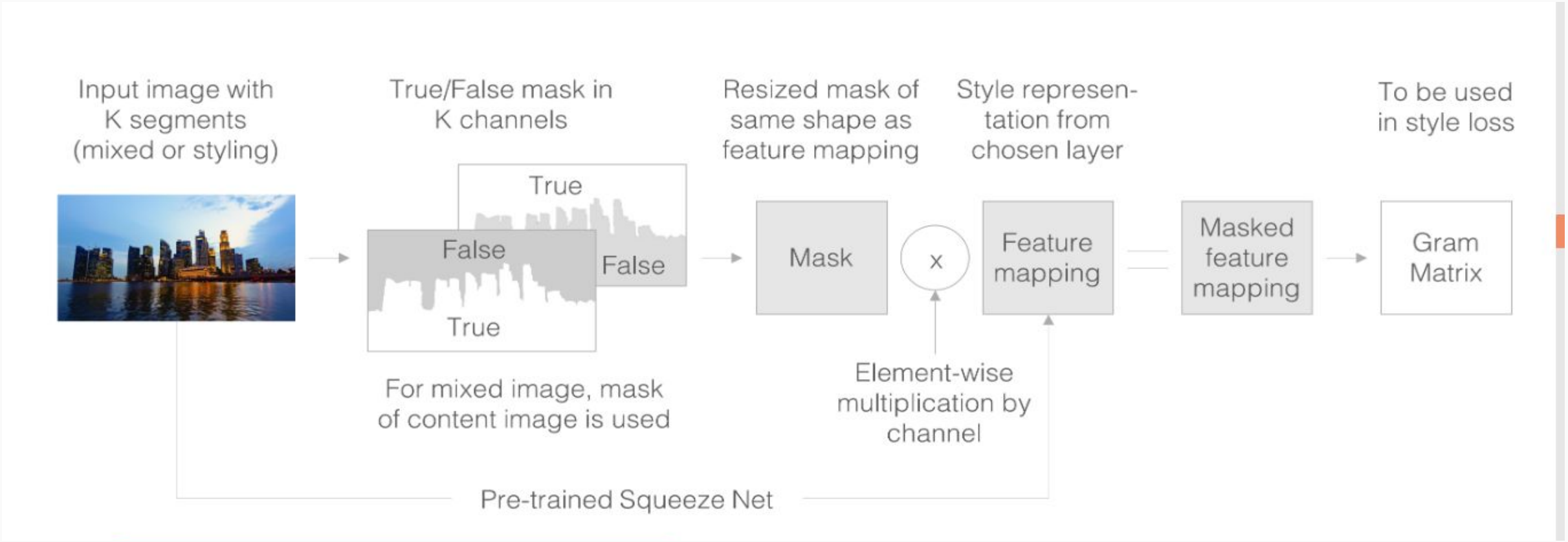
Understanding Semantic Segmentation



(a) Image (b) Unary classifiers (c) Robust P^n CRF (d) Fully connected CRF, MCMC inference, 36 hrs (e) Fully connected CRF, our approach, 0.2 seconds

What's happening?

Steps involved in augmented style loss calculation can be understood in following image (see the paper)



Augmented Style Loss

Style loss updated accordingly with semantic segmentation(called ***Augmented style loss***) can be given by following equation(Ml,c denotes the channel c of the segmentation mask in layer l):

$$\mathcal{L}_{s+}^{\ell} = \sum_{c=1}^{C} \frac{1}{2N_{\ell,c}^2} \sum_{ij} (G_{\ell,c}[O] - G_{\ell,c}[S])_{ij}^2 \quad \text{(3a)}$$

$$F_{\ell,c}[O] = F_{\ell}[O] M_{\ell,c}[I] \quad F_{\ell,c}[S] = F_{\ell}[S] M_{\ell,c}[S] \quad \text{(3b)}$$

Simplified version of the above equation:

$$\mathscr{L}_{\text{style}}^{+} = \sum_{k \in segments} \sum_{l \in layers} \sum_{i,j} (G_k^{\ell} - A_k^{\ell})_{ij}^2 \quad (6)$$

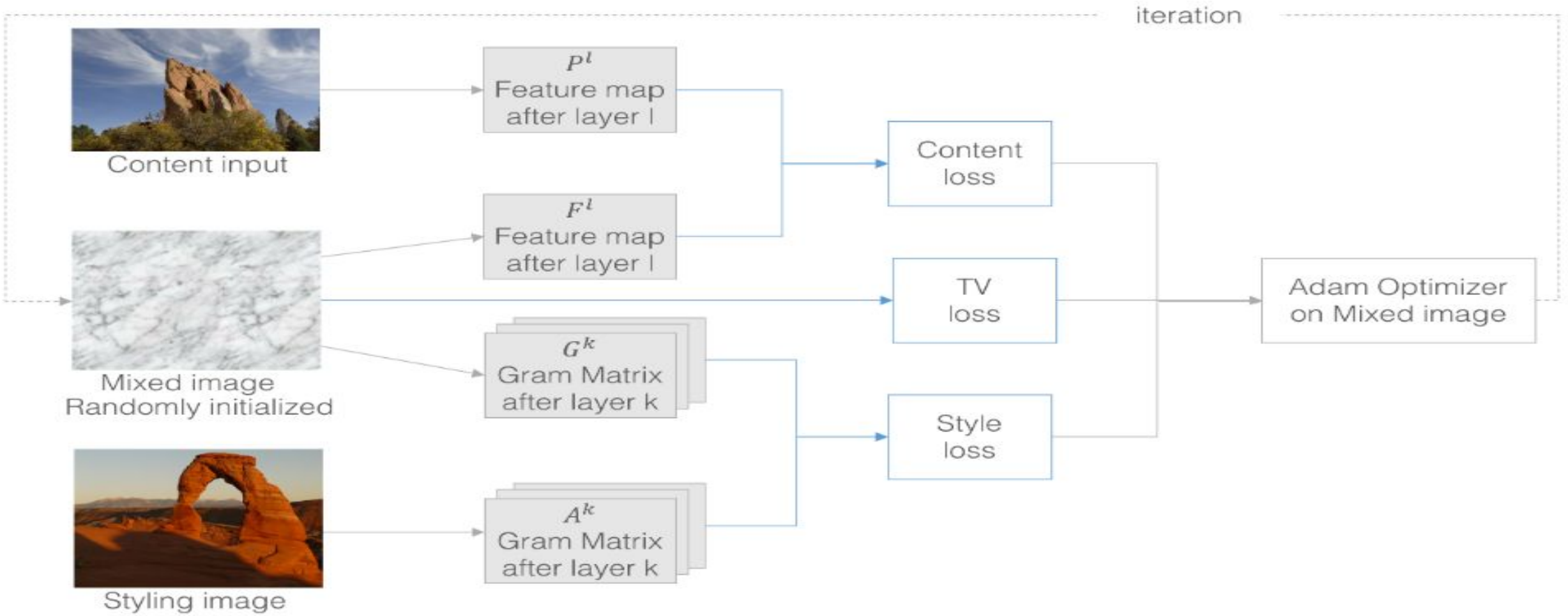
Updated Total loss

Using modified style loss and photorealism regularization total loss can be given by the following equation as given in the paper(**λ is** a weight that controls the photorealism regularization):

$$\mathcal{L}_{\text{total}} = \sum_{l=1}^{L} \alpha_\ell \mathcal{L}_c^\ell + \Gamma \sum_{\ell=1}^{L} \beta_\ell \mathcal{L}_{s+}^\ell + \lambda \mathcal{L}_m$$

Understanding all together

All the concept discussed till now can be wrapped up by the following image from (this) paper(tv loss is photorealism regularization term):



Conclusion

- The paper introduced a deep-learning approach that faithfully transfers style from a reference image for a wide variety of image content.
- It used the Matting Laplacian to constrain the transformation from the input to the output to be locally affine in color-space
- Semantic segmentation further drives more meaningful style transfer yielding satisfying photo-realistic results in a broad variety of scenarios, including transfer of the time of day, weather, season, and artistic edits.

Thanks!